Link and Node Prediction in Metabolic Networks with Probabilistic Logic

Angelika Kimmig and Fabrizio Costa

Departement Computerwetenschappen, K.U. Leuven
Celestijnenlaan 200A - bus 2402, B-3001 Heverlee, Belgium
{angelika.kimmig, fabrizio.costa}@cs.kuleuven.be

Abstract. Information on metabolic processes for hundreds of organisms is available in public databases. However, this information is often incomplete or affected by uncertainty. Systems capable to perform automatic curation of these databases and capable to suggest pathway-holes fillings are therefore needed. Using ProbLog, a simple yet powerful extension of the logic programming language Prolog with independent random variables, we start to investigate two fundamental problems concerning automatic metabolic networks curation, namely *link prediction* and *node prediction*.

1 Introduction

We are nowadays capable of representing organism-wide metabolic processes. In fact there exist collections of metabolic networks for several hundreds of organisms (e.g. the Kyoto Encyclopedia of Genes and Genomes (KEGG) [1] or the BioCyc database [2]) where relations between genes, enzymes, reactions and chemical compounds are available. The knowledge that we have of these relations is however incomplete (most annotation efforts fail to assign functions to 40-60% of the sequences [3]) and is affected by uncertainty (wrong EC number assignment, incomplete annotation (e.g. only one function of a multidomain protein) or nonspecific assignment (e.g. to a protein family)). Systems capable to perform automatic curation of these databases and capable to suggest pathway-holes fillings are therefore needed. For this purpose one can make use of information on related organisms and use evidence based not exclusively on homology searches, but also on genomic and/or functional context. This raises the problem of how to integrate heterogeneous and uncertain sources of information in a principled way. Although systems for reconstructing pathways from relevant gene sets [4] and filling pathway-holes [5] are known in literature, they do not offer sufficient flexibility when new additional sources of information become available or, more importantly, in case one needs to change the set of queries involved in the solution of a specific task.

We have studied an approach that satisfies these flexibility requirements by representing metabolic networks in a probabilistic logical framework. In this way background knowledge affected by uncertainty can be easily included, and we can

obtain an answer to several key questions performing probabilistic inference in a principled manner. More specifically, we use ProbLog [6], a simple yet powerful extension of the logic programming language Prolog with independent random variables in the form of *probabilistic facts*.

In this work we start to investigate some fundamental problems concerning automatic metabolic networks curation, namely: 1) *link prediction*, i.e. the correction of the link strength between a gene and an enzyme, and 2) *node prediction*, that is, whether the existence of a certain enzyme (and hence of an unknown gene) has to be hypothesized in order to maintain the contiguity of a pathway.

2 The Probabilistic Logic Environment: ProbLog

In contrast to propositional graphical models (such as Bayesian Networks), connections between random variables in ProbLog can be specified on the first order level, thus avoiding the need of explicitly grounding all information a priori, achieving therefore a higher abstraction and flexibility in the queries specification.

More formally, a *ProbLog program* T consists of a set of labeled facts $p_i :: c_i$ together with a set of definite clauses encoding *background knowledge* (BK). Each ground instance of such a fact c_i is true with probability p_i, where all probabilities are assumed mutually independent. The program thus naturally defines a probability distribution

$$P(L|T) = \prod_{c_i \in L} p_i \prod_{c_i \in L_T \setminus L} (1 - p_i)$$

over logic programs $L \subseteq L_T = \{c_1, \cdots, c_n\}$. The *success probability* of query q is then defined as

$$P_s(q|T) = \sum_{L \subseteq L_T} P(q|L) \cdot P(L|T) \tag{1}$$

where $P(q|L) = 1$ if there exists a θ such that $L \cup BK \models q\theta$, $P(q|L) = 0$ otherwise. It thus corresponds to the probability that q is *provable* in a randomly sampled logic program. To calculate success probabilities, ProbLog 1) constructs a DNF formula representing all proofs of the query, and 2) uses Binary Decision Diagrams (BDDs) [7] to efficiently calculate the probability of this formula being true in a randomly sampled program. The probability of a DNF formula cannot be obtained directly from the probabilities of the different proofs, as each possible world can allow for multiple proofs. This problem is also known as the *disjoint-sum-problem* or the two-terminal network reliability problem, which is #P-complete [8]. BDDs offer a way to tackle the problem whithout the need to enumerate all possible worlds by compactly representing a Boolean formula as an acyclic directed graph. A DNF formula could naively be encoded as a full Boolean decision tree where each layer corresponds to one probabilistic fact and each leaf is labeled with the truth value of the query in the world given by the

truth value assignments on the path to this leaf. Using BDDs is similar in spirit, with two important differences. First, redundancies in the tree are exploited to obtain compact representations, and second, BDDs are built by combining BDDs for subformulas, thus avoiding the need to construct the entire tree. Probabilities can then be calculated by a single bottom-up pass through the final structure; we refer to [6] for more details.

3 Method

3.1 Metabolic Networks

We represent the knowledge about metabolic networks in a probabilistic logical framework. To this end, we identify the main entities involved in the problem and encode all relations between them quantifying the uncertainty of each relation with an associated probability value (see Figure 1). The entities that we consider are: organisms, genes, enzymes, reactions, compounds (also called metabolites) and pathways; the relationships considered are: organisms are phylogenetically related to other organisms; enzymes are related to enzymes in the enzyme functional hierarchy given by the Enzyme Commission number (EC number) system [9]; genes are related to genes via the ortholog relationship (see further in the text); genes are related to the organisms they are part of; reactions are related to the compounds they require as substrate and to those that are produced; genes are related to the enzymatic function of the protein that they code for; enzymes are related to the reactions they catalyze; and finally pathways are collections of related reactions. Currently only the gene-enzyme relation is treated probabilistically while all the other relations are assumed to be known with certainty and are derived from the KEGG Database [1]. Note that in principle all relations are of the type many-to-many although in practice a gene is almost always associated to a single enzyme, which in turn catalyzes almost always a single reaction (see Figure 1). Informally, a metabolic network contains information on the set of genes that belong to specific organisms and how these code for proteins, called enzymes, that are responsible for specific reactions involving the transformation of one compound into another. An organism is thus capable to perform certain related sets of reactions (semantically grouped under a single pathway concept) in order to produce and transform sets of metabolites, only if the organism can express the enzymes needed to catalyze those reactions.

3.2 Learning Task

Given the metabolic information about a set of organisms we identify two main problems of interest relevant for the concept of automatic network curation: 1) *link prediction*, where we estimate the probability associated to a given set of relations on the basis of an initial guess, so to increase the consistency with respect to the information on related organisms; and 2) *node prediction*, where we introduce specific nodes in order to best fill gaps in the pathway of interest.

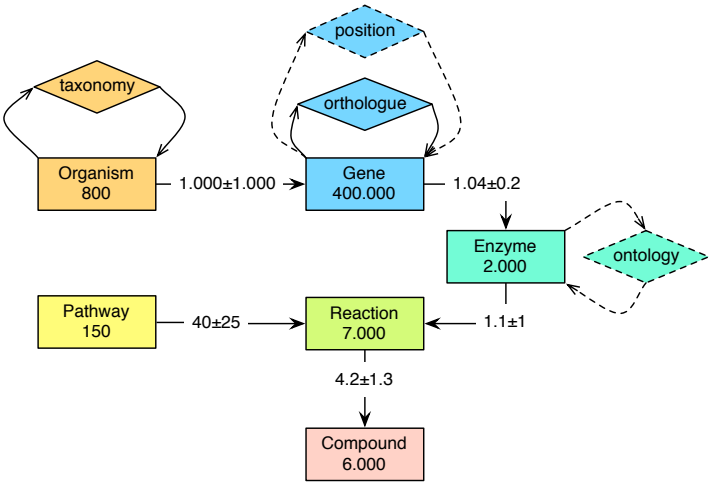

Fig. 1. Part of KEGG metabolic network used. The number in the node shape is the cardinality of the element set. The number on the edge is the average ± standard deviation number of relations between the element at the starting endpoint and the elements at the final endpoint of the edge. Dashed elements represent information present in KEGG but not currently used.

More in detail, we work in the following setting: we are given information about a new organism consisting of a set of genes and their associated functions (i.e. the enzyme they code for); this information is understood affected by uncertainty and is a preliminary approximation that needs to be refined to increase consistency; as background knowledge we are given information on the metabolic network for a large set of organisms; furthermore we are given two similarity notions: the first one is between the test organism and other organisms (obtained from the phylogenetic tree) and the second is between the genes in the test organism and genes in other organisms. This latter information is available in KEGG and is obtained via an heuristic method that determines an ortholog cluster identifier in a bottom-up approach. In this method, each gene subgroup is considered as a representative gene and the correspondence is computed using bi-directional best hit (BBH) relations obtained from the KEGG SSDB database which stores all-vs-all Smith-Waterman similarity scores [10]. For efficiency reasons, both similarity scores are currently thresholded and binarized: in practice, two genes are linked via the ortholog relation only when each one is ranked in the top k most similar genes of the other and when the similarity between the two exceeds a pre-specified threshold.

Link prediction task: In order to re-estimate the strength of a gene-enzyme relation, we consider evidence coming from different types of substructures in the network (see Figure 2 left and Figure 3) and use the ProbLog inference engine to compute the associated probability value. In principle we prefer evidence coming

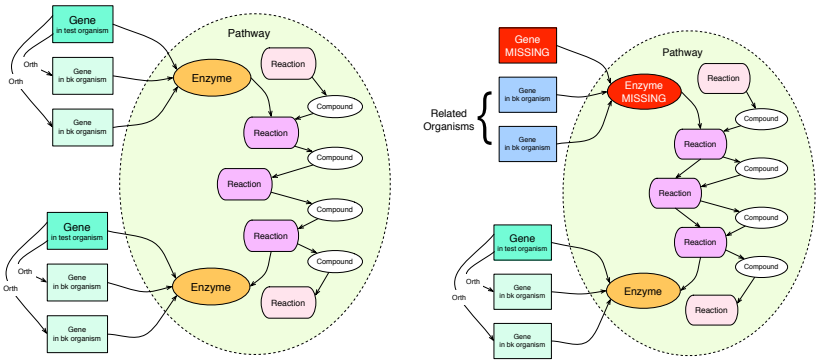

Fig. 2. Graphical representation of the portion of metabolic network used to obtain evidence for the link prediction task (left) and the node prediction task (right).

from more complex substructures but in practice this information is not always available. The reason for this phenomenon is the partial knowledge that we have of the metabolic network: a) not all genes of a test organism have an initial associated function; b) not all genes have known orthologs; c) not all reactions are known in a given pathway. If the database does not contain information to completely match a complex query against, this query will simply fail as it cannot be proven. Hence, it does not provide any information, or, in other words, contributes a probability of 0. In these cases we resort to increasingly simpler queries in a fashion similar in spirit to the interpolation techniques employed in computational linguistic[1]. Finally we integrate information coming from the increasingly complex queries via linear model whose coefficients are learned under a supervised scheme.

In order of complexity we consider: 1) evidence of the strength of gene-enzyme relation either known a-priori or computed by a predictive system (Figure 3 left), this corresponds to the initial estimate embodied as a link between the gene and the enzyme; 2) evidence coming from paths, that is, the probability of a path that involves the gene-enzyme link under consideration (Figure 3 middle); and 3) evidence coming from a complex subgraph, that is the probability of a network portion that involves both the gene-enzyme link and links of ortholog genes in related organisms (Figure 3 right). ProbLog allows us to specify the characteristics of these substructures at an intensional level. In particular we require 2) to be a path that traverses in order the following selected types of entities: gene, enzyme, reaction, compound, (reaction-compound)*, reaction, enzyme, gene. The intended meaning of the star notation here is that the path is only allowed to follow further reaction-compound links if the current reaction does not have an

[1] When employing *n-gram* models, a common practice is to assess the probability of complex n-grams using the frequency counts of smaller n-grams that are more likely to occur in (small) datasets.

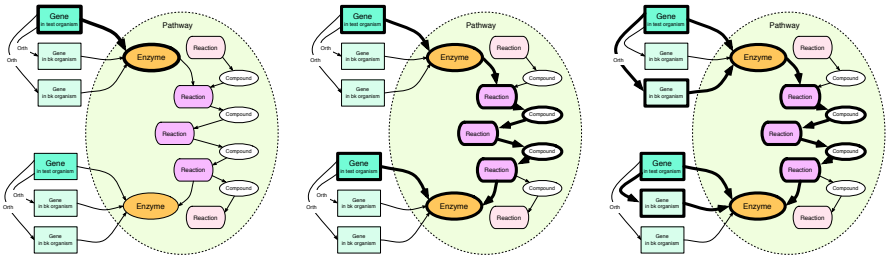

Fig. 3. Graphical representation of the types of substructures used to obtain evidence for the link prediction task (marked in bold): single gene-enzyme edge (left), path between two genes (middle) and subgraph involving ortholog genes (right).

enzyme associated in the database. This latter condition is motivated by both computational efficiency issues (i.e. we do not consider all possible paths but only the shortest ones) and the desire to favor paths that make use of information relevant to the test organism. In words: we consider linear chains that originate in one gene of the test organism and end up in another gene of the same organism traversing the enzyme-reaction network relevant to a specific pathway. The subgraph for case 3) is obtained considering paths of type 2 with the addition of two extra paths that originate from genes in the test organism, traverses ortholog genes and end up in the enzymes of interest at both ends of the original path of type 2. The ratio here is to prefer evidence that is consistent with the information on similar genes in different organisms.

Node prediction task: Here we compute the probability of an enzyme and adapt the structures we use to provide evidence in the following way (see Figure 2 right): first of all, note that we cannot consider structures of type 1), that is an a-priori estimate as we work precisely under the assumption that no information is known on the existence of a gene associated to a specific enzymatic activity; instead we consider the average association strength of given enzyme with any known gene present in related organisms; for the more complex queries we consider paths similar to those of type 2) where the initial gene-enzyme link is removed and is substituted by a gene in some other related organism, but where we still require the path to end in a gene that is known to belong to the test organism; finally for the most complex query we consider subgraphs similar to 3) where the initial gene-ortholog gene-enzyme chain is replaced by a gene-enzyme relation between a gene belonging to a related organism; for the other endpoint of the path we use the information available about genes that are orthologs of test genes.

Learning task: In both the link and node prediction setting we introduce queries that can be answered in multiple ways (e.g. there is more than one path that starts from a given initial gene and ends in another gene of the same organism). Each solution comes equipped with an associated probability of being true and each contributes evidence to the overall probability to satisfy the query

predicate. Since the various solutions are not independent we cannot derive the final probability simply by summing up all the returned probabilities. This is an instance of the *disjoint-sum-problem*, which in ProbLog is tackled by resorting to BDDs as explained in Section 2.

Finally we use the results returned for each different query type to answer the main questions: what is the probability of a specific gene of a test organism to be associated to a specific enzyme? or what is the probability of a specific enzyme to belong to a pathway for a given test organism? We compute these probabilities via a linear model whose parameters are learned using ProbLog's gradient-descent approach to parameter learning [11]. Given a set of queries with associated target probabilities, this method uses standard gradient descent to minimize the *mean squared error* (MSE) on the training data. To do so, it exploits the BDDs used in ProbLog inference to also calculate the gradient.

The idea behind the linear model is to learn how to adapt to the level of missing information in the network: when predicting the association strength with an enzyme that is embedded in a network region where few reactions are known it is better to trust the prior estimate with respect to more complex queries since they will mainly fail over the poorly connected reaction network; analogously when ortholog genes are known for a given enzyme, the evidence from the more complex queries becomes compelling. In summary, we adapt to the unknown local quality of the network by learning the relative importance of each query for the final answer, and we do this by training a model on related organisms.

4 Experimental Setup

4.1 Noise Model

Instead of working with a specific gene function predictor we study the curation/reconstruction capacity of the proposed system perturbing the knowledge of the true function of a gene in a specific and controlled way. Since the enzymatic functions can be arranged in a hierarchical ontology [9] we posit that we can relate the topological distance in the ontology tree to the functional distance, i.e. the closer two enzyme nodes are in the hierarchy the more similar their functions. Under this assumption we build a noise model described by the following parameters: 1) s fraction of affected genes; 2) k number of noisy gene-enzyme links added per gene; 3) σ_{EC} parameter controlling the size of the neighborhood where to randomly sample the additional noisy gene-enzyme links; 4) σ_N parameter controlling the quantity of noise added to the gene-enzyme relationship probability estimate. We then proceed as follows (see Figure 4): given an organism we select a fraction s of its genes; for each gene we add k extra links to randomly sampled *nearby* enzymes; here the informal notion of a metric is formally defined as a normal probability distribution (of selecting an enzyme) $N(0, \sigma_{EC})$ and support over the topological distance induced by the ontology (i.e. the length of the shortest path between the leafs containing the two enzymes in the tree structured ontology); finally the strength of the link between the gene

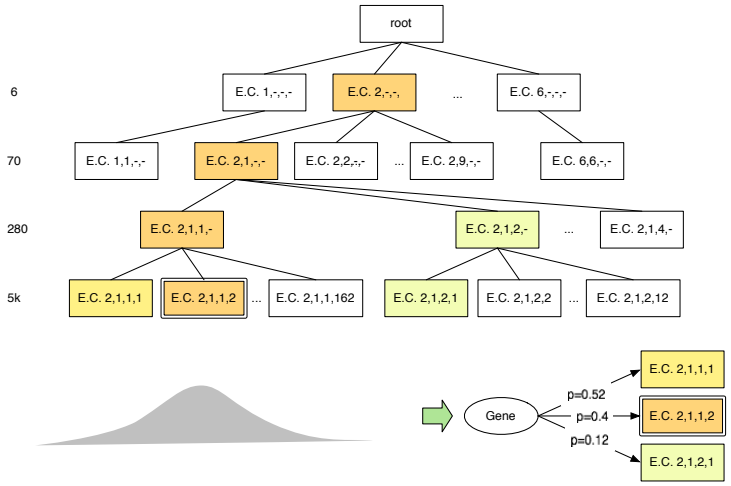

Fig. 4. Noise model: the E.C. hierarchy induced metric notion (i.e. topological distance between nodes) is used as support for the perturbed enzymatic function. The hypothetical true enzyme is marked with a double line. In the example a gene is associated to an incorrect enzymatic activity with probability 0.52 and to the correct one with probability 0.4.

and the randomly selected enzyme is computed as the probability of selecting the enzyme with additional $N(0, \sigma_N)$ noise: in this way enzymes that are less related to (i.e. more distant from) the true enzymatic function of the original gene receive on average a smaller probability.

4.2 Experimental Results

In the experiments reported here, we focus on the Pyruvate metabolism pathway for the Escherichia coli UTI89 test organism. We perturb the true relationships with k=5 extra links for $s = 50\%$ of genes. The probability estimate of the gene-enzyme relationship receives additional noise from $N(0, \frac{1}{8})$.

We use default settings in our experiments and run learning for at most 50 iterations, stopping earlier if the MSE on the training data does not change between two successive iterations. Training data is generated from the other organisms with the same parent in the organism hierarchy as the test organism, and target probabilities are set to 1.0 for positive and 0.0 for negative examples, respectively.

In the link prediction setting, positive examples are real gene-enzyme links, while negative ones are the ones added by the noise model where no real one is known between these entities. We use the three queries depicted in Figure 3. We measure the area under the precision-recall curve.

When using the initial (perturbed) estimate for the gene-enzyme link we achieve an AUCPR of 0.69. If we use only the most complex query (type 3) we

increase to 0.74, but when we learn the logistic model over all queries we achieve 0.80. Note that simply learning a fixed mixture of experts for the whole organism (i.e. not modeling the dependency on the enzyme) we do not improve over the initial 0.69 result as for this particular test organism, it is better to resort on average to the most simple query.

In the node prediction experiment, we adopt an enzyme level leave-one-out design. From the background knowledge we retrieve all the enzymes that do not have an associated gene in the test organism. We remove all enzymes in turn and we measure the precision at one, that is the fraction of times that the missing enzyme is ranked in first position as the most probable among all the missing enzymes.

The set of training examples is the set of all pairs of training organisms (as before) and enzymes appearing in the pathway for organisms different from the test organism. Such a pair is considered positive if the enzyme appears in the organism's pathway, and negative else.

We use the query described in Section 3 both with and without ortholog information, as well as a basic query that predicts each enzyme with the average probability of a gene-enzyme link involving this enzyme in one of the training organisms. In this experiment we achieve a precision at one of 0.66 over 35 possible enzymes (i.e. the baseline random guessing precision at one would be 0.03).

5 Conclusions

We have started tackling the problem of automatic network curation by employing the ProbLog probabilistic logic framework. This choice has allowed us to: a) represent the knowledge about the metabolic network even when affected by uncertainty, and b) express complex queries to extract evidence for the presence of missing links or nodes in an abstract and flexible way. Initial experimental evidence shows that we can effectively recover missing or inconsistent information. Future work includes the integration of concrete gene function predictor and the development of novel queries that make use of additional sources of information such as the gene position in the genome or the co-expression of genes in the same pathway from medical literature abstract analysis.

Acknowledgments

A. Kimmig is supported by the Research Foundation-Flanders (FWO-Vlaanderen). This work is partially supported by the GOA project 2008/08 Probabilistic Logic Learning and by the European Commission under the 7th Framework Programme, contract no. BISON-211898.

References

1. Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., Hirakawa, M.: Kegg for representation and analysis of molecular networks involving diseases and drugs. Nucleic Acids Research **38**(Database issue) (2010) D355
2. Karp, P., Ouzounis, C., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahren, D., Tsoka, S., Darzentas, N., Kunin, V., Lopez-Bigas, N.: Expansion of the biocyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Research **19** (2005) 6083–89
3. Pouliot, Y., Karp, P.: A survey of orphan enzyme activities. BMC bioinformatics **8**(1) (2007) 244
4. Moriya, Y., Itoh, M., Okuda, S., Yoshizawa, A., Kanehisa, M.: Kaas: an automatic genome annotation and pathway reconstruction server. Nucleic Acids Research **35**(Web Server issue) (2007) W182
5. Green, M., Karp, P.: A bayesian method for identifying missing enzymes in predicted metabolic pathway databases. BMC bioinformatics **5**(1) (2004) 76
6. Kimmig, A., Santos Costa, V., Rocha, R., Demoen, B., De Raedt, L.: On the Efficient Execution of ProbLog Programs. In de la Banda, M.G., Pontelli, E., eds.: International Conference on Logic Programming. Number 5366 in LNCS, Springer (December 2008) 175–189
7. Bryant, R.E.: Graph-based algorithms for boolean function manipulation. IEEE Trans. Computers **35**(8) (1986) 677–691
8. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM Journal on Computing **8**(3) (1979) 410–421
9. Webb, E.C.: Enzyme nomenclature 1992: recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes. San Diego: Published for the International Union of Biochemistry and Molecular Biology by Academic Press (1992)
10. Moriya, Y., Katayama, T., Nakaya, A., Itoh, M., Yoshizawa, A., Okuda, S., Kanehisa, M.: Automatic generation of kegg oc (ortholog cluster) and its assignment to draft genomes. International Conference on Genome Informatics (2004)
11. Gutmann, B., Kimmig, A., Kersting, K., De Raedt, L.: Parameter learning in probabilistic databases: A least squares approach. In Daelemans, W., Goethals, B., Morik, K., eds.: European Conference on Machine Learning. Volume 5211 of LNCS., Springer (2008) 473–488